

PAT-NO: JP410083326A
DOCUMENT-IDENTIFIER: JP 10083326 A
TITLE: TRACE DATA COLLECTING METHOD AND COMPUTER READABLE CODE
PUBN-DATE: March 31, 1998

INVENTOR-INFORMATION:

NAME	COUNTRY
ANTHONY, H RINDOSEI	

ASSIGNEE-INFORMATION:

NAME	COUNTRY
INTERNATL BUSINESS MACH CORP	N/A

APPL-NO: JP09106126
APPL-DATE: April 23, 1997

INT-CL (IPC): G06F011/28

ABSTRACT:

PROBLEM TO BE SOLVED: To provide a technique enabling obtaining a trace point set in connection with the data constituting element of an object oriented program on the purpose of debugging.

SOLUTION: In an object including data related with which tracing is necessitated, a new method is generated or an existent method is corrected. The new or corrected method is capable of interpreting a message desiring to access designated data to start, correct and finish tracing corresponding to the kind of trying access and another condition set by a developer. Tracing is controlled by sending a proper message to a trace manager. Reading and writing operation is stored by a newly generated access method executing access to a necessitated data constituting element.

COPYRIGHT: (C)1998,JPO

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-83326

(43) 公開日 平成10年(1998) 3月31日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 11/28	3 1 0		G 0 6 F 11/28	3 1 0 B

審査請求 未請求 請求項の数14 O L (全 14 頁)

(21) 出願番号 特願平9-106126

(22) 出願日 平成9年(1997) 4月23日

(31) 優先権主張番号 08/640613

(32) 優先日 1996年5月1日

(33) 優先権主張国 米国 (US)

(71) 出願人 390009531

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(72) 発明者 アンソニー・エイチ・リンドセイ

アメリカ合衆国27526、ノースカロライナ
州 フグエイ ヴァリナ パースクロフト
ブレイス 4008

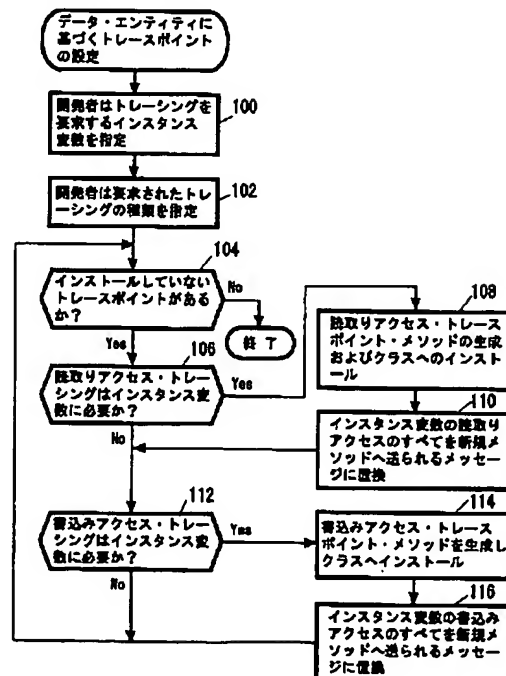
(74) 代理人 弁理士 坂口 博 (外1名)

(54) 【発明の名称】 トレース・データ収集方法およびコンピュータ読み取り可能コード

(57) 【要約】

【課題】デバッグを目的としたオブジェクト指向プログラムのデータ構成要素に関連して設定されるトレースポイントを可能とする技術を提供する。

【解決手段】どのトレーシングが必要であるかに関連したデータを含むオブジェクトにおいて、新たなメソッドを生成するか、もしくは既存のメソッドを修正する。新規もしくは修正されたメソッドは、指定されたデータにアクセスしようとするメッセージを解釈し、試みられているアクセスの種類および開発者によって設定された他の条件に応じて、トレーシングを開始、修正、または終了することができる。トレーシングは、適当なメッセージをトレース・マネージャへ送ることによって制御される。読み取りまたは書き込み操作は、必要とされるデータ構成要素へのアクセスを実行する新たに生成されたアクセス・メソッドによって保存される。



【特許請求の範囲】

【請求項1】オブジェクト指向プログラムのデータ構成要素に対するアクセスに基づく、前記オブジェクト指向プログラムのデバッグ中にトレース情報を収集することを可能とするデータ処理環境でのプロセッサ上での実行のためのコンピュータ読み取り可能コードであつて、

どのデータ構成要素に対するアクセスがトレース情報の収集を開始するのに必要であるかに関連する前記オブジェクト指向プログラムのデータ構成要素を、ユーザが指定することを可能とする第1のサブプロセスと、指定されたデータ構成要素へ送られるメッセージを送ることによってトリガーされたトレース情報の収集を開始する第2のサブプロセスとを備えることを特徴とするコンピュータ読み取り可能コード。

【請求項2】前記第1のサブプロセスは、前記指定されたデータ構成要素にアクセスすることに基づいた前記トレース情報の回収を制御するパラメータを、さらにユーザが指定できるようにすることを特徴とする請求項1に記載のコンピュータ読み取り可能コード。

【請求項3】前記パラメータの一つは、どのトレースされた情報の収集が終了するかに対する事像であり、さらに前記終了の事像の発生に対するトレース情報の回収を終わらせる第3のサブプロセスを備えることを特徴とする請求項2に記載のコンピュータ読み取り可能コード。

【請求項4】前記第1のサブプロセスは、前記指定されたデータ構成要素に対する少なくとも一つのアクセス・メソッドを生成することと、前記オブジェクト指向プログラムのメソッドにある前記指定されたデータ構成要素に対する直接参照のすべてを、前記アクセス・メソッドに送られるメッセージに置き換えることとを、さらに含むことを特徴とする請求項1に記載のコンピュータ読み取り可能コード。

【請求項5】前記指定されたデータ構成要素は、オブジェクト指向プログラムのオブジェクトのインスタンス変数であることを特徴とする請求項1に記載のコンピュータ読み取り可能コード。

【請求項6】前記アクセス・メソッドは、トレース情報の収集を制御することを特徴とする請求項4に記載のコンピュータ読み取り可能コード。

【請求項7】前記第1のサブプロセスは、読み取りトレースポイント・アクセス・メソッドおよび書き込みアクセス・トレースポイントを生成し、さらに前記オブジェクト指向プログラムによって発生した前記指定されたデータ構成要素に対する読み取りの試みのすべては、読み取りトレースポイント・アクセス・メソッドに送られ、また前記オブジェクト指向プログラムによって発生した前記指定されたデータ構成要素に対する書き込みの試みのすべては、書き込みトレースポイント・アクセス・メソッドに送られることを特徴とする請求項1に記載のコ

ンピュータ読み取り可能コード。

【請求項8】前記アクセス・メソッドは、前記指定されたデータ構成要素にアクセスするメッセージを受け取った後に前記トレース情報の収集を開始させることを特徴とする請求項4に記載のコンピュータ読み取り可能コード。

【請求項9】アクセス・メッセージを受け取った後に要求されたアクセス機能を実行する前記アクセス・メソッドを生成する第4のサブプロセスと、

前記アクセス・メソッドが前記要求されたアクセス機能を実行した後に、トレース・データの収集を修正する第5のサブプロセスとを、さらに備えることを特徴とする請求項8に記載のコンピュータ読み取り可能コード。

【請求項10】少なくともプロセッサとメモリとが含まれるコンピュータ環境下で、デバッグを受けているオブジェクト指向プログラムのデータ構成要素へのアクセスに基づいて収集されるべきトレース・データを生ずるシステムであつて、

前記データ構成要素へのアクセスの試みがトレース情報の収集を開始するために要求される前記オブジェクト指向プログラムの少なくとも一つのデータ構成要素を指定する手段と、

指定された前記データ構成要素がアクセスされた場合に、トレース・データの収集をトリガーする手段とを備えたことを特徴とするシステム。

【請求項11】トレーシング・パラメータを指定する手段を、さらに有することを特徴とする請求項10に記載のシステム。

【請求項12】デバッグしているオブジェクト指向プログラムのデータ構成要素がアクセスされた場合のトレース・データ収集方法であつて、アクセスの試みがトレーシングに影響を及ぼす前記オブジェクト指向プログラムのデータ構成要素を指定する段階と、

指定されたデータ構成要素に対するアクセスの試みがデバッグ中に送られる少なくとも一つのアクセス・メソッドを生成する段階と、

前記アクセス・メソッドが前記指定されたデータ構成要素に対するアクセスの試みを受けた場合にトレース・データの収集をトリガーする段階とを有することを特徴とするトレース・データ収集方法。

【請求項13】前記少なくとも一つのアクセス・メソッドを生成する段階は、前記指定されたデータ構成要素を要求するすべての読み取りアクセスと、前記指定されたデータ構成要素を要求するすべての書き込みアクセスとを生成することを特徴とする請求項12に記載のトレース・データ収集方法。

【請求項14】前記アクセス・メソッドが前記アクセスの試みを受けた後、前記指定されたデータ構成要素にアクセスする段階と、

前記指定されたデータ構成要素がアクセスされた後に前記トレース・データの収集を修正する段階とを、さらに有することを特徴とする請求項12に記載のトレース・データ収集方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、オブジェクト指向性ソフトウェア・アプリケーションをデバッグするための方法に関する。特に、本発明はソフトウェア・アプリケーションをデバッグするデバッガにおけるデータ・トリガー・トレース・サポートを提供する。

【0002】

【従来の技術】ソフトウェア・プログラムを開発するプロセスにおいて一つのクリティカルで、かつ見落とされがちな構成要素は、ソフトウェア・プログラムからのエラーの排除である。エラーを取り除くプロセスは、デバッグとして知られている。一人あるいはそれ以上の数のプログラム開発者がソフトウェア・プログラムのためのプログラム用コードを記述し、さらにコード・ファイルを構築することによってソフトウェア・プログラムの実行可能なバージョンを作る。その後、ソフトウェアプログラムにどのようなエラーが存在するかを判断するために、プログラムのテストを行わなければならない。次に、プログラムのデバッグを行い、エラーを少なくするか、あるいは取り除かなければならない。業務用ソフトウェアに関して言えば、ソフトウェアが市場に出回る前にほとんどのエラーまたはバグを見つけ出し、それらを取り除いておくことが絶対必要である。デバッグを行うプロセスは、かなりの費用がかさむものであり、また多くの時間を費やすにちがいない。したがって、コードのデバッグを支援し、かつデバッグのサイクル・タイムを減少させるソフトウェア・ツールおよび技術が開発されている。

【0003】市販のソフトウェアアプリケーションで発見されるエラーまたはバグに関して言えば、時間がきわめて重要である。ソフトウェアの製造元は、バグについての顧客の不平に迅速に対応するツールを持たなければならない。またバグが生じたソフトウェアを素早く修復またはデバッグしなければならない。ソフトウェアのエラーまたはバグを2通りのタイプに分類することができる。第1のタイプは論理エラーであり、ソフトウェア開発者によってプログラムに加えられた論理文または命令に、必要とする機能がプログラムの実行中に実行されないか、あるいは適切に実行されない等の欠陥が生ずる。一般に、論理エラーによって開発者が望まないプログラム中の機能またはコマンドが実行される。エラーの第2のタイプは、プログラムによって維持または使用されているデータに関するもので、データが何らかの理由で壊れた場合に起こる。

【0004】多数のデバッグ機能および技術が現在

入手可能なデバッグ・ソフトウェア・ツールに現在使われている。既知の方法には、テストポイントと呼ばれる一組の方法があり、ブレークポイント、トレースポイント、ウォッチポイント等の設定が含まれる。しかし、トレースポイントの使用は多少なりとも限定されている。デバッグしているプログラムで特定論理ユニットの実行に対するトレース情報の作成をトリガーするようにトレースポイントを設定するツールが存在する。その一方で、特定のデータ要素あるいはエンティティ、例えばデバッグされるプログラム内で、あるいはデバッグされているプログラムによって利用されるSmalltalk言語プログラムの変数を基準としてトレーシングがトリガーされるようにしてトレースポイントの設定を行う同様の機能が存在する。現行論理ベースのトレース機能は、データの動きに関する情報を作成することができる。しかし、Smalltalkメソッドのような論理ユニット実行上で起こるデータの動きのコンテキスト内のみである。さらに、論理ユニットの実行による影響を受けたデータ構成要素のすべてに関連して情報が作成される。情報がシングル・データ・エンティティに関連してソートされている場合、論理に基づいたトレースポイント手順は時間処理の観点から言えばコストがかかる。なぜなら、影響を受けたデータ構成要素のすべてに関連した情報が集められ、開発者の時間の観点から言えば、開発者は必要なデータに属する情報を見つけ出すために影響を受けたデータ構成成分のすべてに関係した情報をソートしなければならない。論理ユニットの実行中にデータ構成要素に関連して情報が得られるように、特定のデータ構成要素にもとづいたトレース・データの生成を引き起こす能力は、プログラムのデバッグにおいて開発者にとって貴重なツールになるかもしれない。

【0005】最近、ソフトウェア産業においてソートの革命が起こっている。最近に至るまで、ほとんどのソフトウェア・プログラムはCおよびCOBOL等の手順言語で書かれていた。したがって、デバッグ・ツールおよび方法は手順プログラムのバグを取るように設計されている。しかし、オブジェクト指向言語が出現することによってパラダイムの変更が生じた。オブジェクト指向言語は、開発者によって記述され、またランタイム時にコンピュータによって実行されるという点で基本的に異なる。2種類の主要なオブジェクト指向言語、C++およびSmalltalkは、これらの言語で書かれたオブジェクト指向プログラムのオブジェクトの中に実際に格納される。SmalltalkではデータをSmalltalkオブジェクトのインスタンス変数およびSmalltalkクラスのクラス変数またはクラス・インスタンス変数に格納し、またC++ではデータをC++オブジェクトの変数に格納する。したがって、オブジェクト指向プログラムによってデバッグ・ツールに対する新たな挑戦の道が開かれた。

【0006】米国特許第5,093,914号(Coplienら)はオブジェクト指向プログラムのデバッグのブレークポイントを用いる方法を開示している。ブレークポイントはテスト・ポイントの一つのタイプであることから、この特許に対する関心が持たれている。しかし、上記米国特許はトレースポイントを扱うものではなく、またオブジェクト指向プログラムのデータ構成要素に関連したトレース機能が欠如しているという問題を取り扱ったり、あるいはその解決策を提供するものではない。

【0007】1992年にリリースされたIBMのクロス・システム・プロダクト(CSP)第4.1版は、手順プログラムをデバッグするオブジェクト指向解決法を提供する。しかし、そのプロダクトは論理ユニットに関連したトレースポイントを利用する一方で、手順プログラムのデータ構成要素に関連したトレースポイントの設定を有効とするものではない。

【0008】したがって、指定されたデータ構成要素に特異的なトレース情報を収集するためにデバッグを受けているオブジェクト指向プログラム内の特定のデータ構成要素のためのトレースポイントを開発者が設定することを可能とするデバッグ・ツールが求められている。

【0009】

【発明が解決しようとする課題】本発明の目的は、特定のデータ構成要素のアクセスまたはそれに対する修正に基づいて、トレーシングを引き起こすトレースポイントによってオブジェクト指向プログラム・コードをデバッグする方法を提供することである。

【0010】本発明の別の目的は、プロセッサのオーバーヘッドを最小限にすることを必要とするオブジェクト指向ソフトウェアの中のデータ構成要素のために、トレースポイントに基づいたデバッグ方法を提供する。

【0011】本発明の他の目的および利点は以下の記載および図面から部分的に説明されよう。また部分的に記載から明かにされるか、あるいは本発明の実施によって学ぶことができよう。

【0012】

【課題を解決するための手段】上記目的を達成するために、また本明細書で大まかに記載されたような本発明の効果にもとづいて、データ・プロセッシング環境におけるプロセッサ上での実行に関した技術について説明する。この技術は、オブジェクト指向プログラムのデータ構成要素へのアクセスに基づいたオブジェクト指向プログラムのデバッグ中におけるトレース情報の収集を可能とするもので、どのデータ構成要素に対するアクセスがトレース情報の収集を開始するのに必要であるかに関連するオブジェクト指向プログラムのデータ構成要素を、ユーザが指定するのを可能とする第1のサブプロセスと、指定されたデータ構成要素へ送られるメッセージを送ることによってトリガーされたトレース情報の収集

を開始する第2のサブプロセスとを有する。第1のサブプロセスは、指定されたデータ構成要素にアクセスすることに基づいたトレース情報の回収を制御するパラメータを、さらにユーザが指定できるようにしてもよい。パラメータの一つは、どのトレースされた情報の収集が終了するかに対する事像としてもよく、さらに上記技術は、終了の事像の発生に対するトレース情報の回収を終わらせる第3のサブプロセスを有するものであってもよい。第1のサブプロセスは、指定されたデータ構成要素に対する少なくとも一つのアクセス・メソッドを生成することと、オブジェクト指向プログラムのメソッドにある指定されたデータ構成要素に対する直接参照のすべてを、新しく生成されたアクセス・メソッドに送られるメッセージに置き換えることとを、さらに含むものであってもよい。第1のサブプロセスは、読み取りトレースポイント・アクセス・メソッドおよび書き込みアクセス・トレースポイントを生成し、さらにオブジェクト指向プログラムによって発生した指定されたデータ構成要素に対する読み取りメッセージのすべては、読み取りトレースポイント・アクセス・メソッドに送られ、またオブジェクト指向プログラムによって発生した指定されたデータ構成要素に対する書き込みメッセージのすべては、書き込みトレースポイント・アクセス・メソッドに送られるようにしてもよい。上記技術は、アクセス・メッセージを受け取った後に要求されたアクセス機能を実行するアクセス・メソッドを生成する第4のサブプロセスと、アクセス・メソッドが要求されたアクセス機能を実行した後に、トレース・データの収集を修正する第5のサブプロセスとを、さらに有するものであってもよい。

【0013】少なくともプロセッサとメモリとが含まれるコンピュータ環境下で、デバッグを受けているオブジェクト指向プログラムのデータ構成要素へのアクセスに基づいて収集されるべきトレース・データを生ずるシステムを提供する。このシステムは、データ構成要素へのアクセスの試みがトレース情報の収集を開始するために要求されるオブジェクト指向プログラムの少なくとも一つのデータ構成要素を指定する手段と、指定されたデータ構成要素がアクセスされた場合に、トレース・データの収集をトリガーする手段とを備える。このシステムは、さらにトレーシング・パラメータを指定する手段を備えるものであってもよい。

【0014】また、デバッグしているオブジェクト指向プログラムのデータ構成要素がアクセスされた場合にトレース・データを収集する方法を提供する。この方法は、アクセスの試みがトレーシングに影響を及ぼす前記オブジェクト指向プログラムのデータ構成要素を指定する段階と、指定されたデータ構成要素に対するアクセスの試みがデバッグ中に送られる少なくとも一つのアクセス・メソッドを生成する段階と、アクセス・メソッドが指定されたデータ構成要素に対するアクセスの試

みを受けた場合にトレース・データの収集をトリガーする段階とを有する。好ましくは、少なくとも一つのアクセス・メソッドを生成する段階は、指定されたデータ構成要素を要求するすべての読み取りアクセスと、指定されたデータ構成用を要求するすべての書き込みアクセスとを生成するものであってもよい。上記方法は、アクセス・メソッドがアクセスの試みを受けた後、指定されたデータ構成要素にアクセスする段階と、指定されたデータ構成要素がアクセスされた後にトレース・データの収集を修正する段階とを、さらに有するものであってもよい。

【0015】以下、図面を参照しながら本発明を詳細に説明する。なお、図中の同一符号は同一構成要素を示す。

【0016】

【発明の実施の形態】本発明は、Smalltalk、C++、SELF等の種々のオブジェクト指向プログラミング言語のいずれにおいても実施することができよう。

【0017】図1は、本発明が適用される典型的なハードウェア環境を示すものである。この図に示す環境は、関連周辺機器を含む代表的な従来のシングル・ユーザ・コンピュータ・ワークステーション10、例えばパーソナル・コンピュータから構成される。

【0018】ワークステーション10は、マイクロプロセッサ12とバス14とを備える。このバス14は、マイクロプロセッサ12とワークステーション10内の構成要素とを既知の技術に従って接続し、かつこれらの間のコミュニケーションを可能とする。ワークステーション10は、一般にバス14を介してマイクロプロセッサ12を一種以上のインタフェース装置、例えばキーボード18、マウス20、および（または）他のインタフェース装置22に接続する。ここで、他のインタフェース装置22の例としては、タッチ・センシティブ・スクリーン、デジタル・ペン入力式パッド等の任意のユーザ・インタフェース装置が挙げられる。

【0019】また、バス14はディスプレイ・アダプタ26を介してLCDスクリーンまたはモニタ等の表示装置24をマイクロプロセッサ12に接続させる。さらに、バス14はマイクロプロセッサ12をROM、RAM等が含まれるメモリ26に接続させる。ワークステーション10は、通信チャンネル30経由で他のコンピューターあるいはコンピューター・ネットワークと通信する。ワークステーション10は、ローカル・エリア・ネットワークまたは広域ネットワークにある他のコンピューターとのクライアント/サーバー構成におけるクライアントとなる。このような構成のすべては、適当な通信ハードウェアおよびソフトウェアと同様に、当業者に既知のものである。

【0020】本発明を具体的に表現するソフトウェア・プログラミング・コードは、一般にワークステーション10のメモリ28に格納される。クライアント/サーバー環境では、そのようなソフトウェア・プログラム・コードを一つのサーバーに関連づけられるメモリに格納してもよい。ソフトウェア・プログラム・コードをデータ・プロセッサ・システムとともに使用される様々な既知のメディア、例えばディスクあるいはCD-ROM上に具体的に表現してもよい。コードをそのようなメディア上で分散してもよく、あるいは一つのコンピュータ・システムのメモリからある種のネットワークを越えて他のコンピュータ・システムへ分散してそのような他のコンピュータ・システムのユーザが使用できるようにしてもよい。ソフトウェア・プログラム・コードを物理的媒体上に具体的に表現するか、あるいはソフトウェア・コードをネットワークを経由して分散することは、既に知られていることからこれ以上の説明は省くこととする。

【0021】以下、本発明の好ましい実施形態例を図2ないし図6を参照しながら説明する。好ましい実施形態例では、本発明をSmalltalkオブジェクト指向言語で実施する。

【0022】オブジェクト指向言語は手順向き言語とは根本的に異なっており、オブジェクト指向言語で記述されたソフトウェア・プログラムの実行はメソッドの実行によって行われる。メソッドはオブジェクトの論理的部分である。メソッドの実行は計算を実行することであり、その後他のオブジェクトへのメッセージ送信を行う。そして次に他のオブジェクトが実行すべき追加の方法を生じ、かつ送信すべきメッセージをさらに生ずる。オブジェクト指向プログラムの特徴および機能は当業者に知られていることからこれ以上の説明は省く。本発明は、トレース情報の収集が要求されることに関連のあるインスタンス変数を持つオブジェクト、あるいはクラス変数またはクラス・インスタンス変数を持つクラスへのメソッドの追加に依存する。デバッグ・ツールの残りの部分については、オブジェクト指向言語で記述される必要性、あるいは実行される必要性はない。本発明によって、オブジェクト指向言語で記述されたプログラムのデータのデータ・エンタイトルに関するトレースポイントの設定が可能となる。

【0023】トレースポイントは、デバッグ・ツールに対して全体のトレース状態を変更する専用のデバッグ・ツールであるテスト・ポイントの一種である。本発明は、オブジェクト指向プログラムの指定されたデータ構成要素にアクセス（読み取りまたは書き込み）にしたがって変更すべき全体のトレース状態を可能とすることによってトレース情報の収集を制御する。Smalltalkでは、トレースポイントを、Smalltalkのデータ実態であるインスタンス変数、クラス変数、またはクラス・インスタンス変数（以下、一括して

インスタンス変数とする)に対応付けることによってトレースをトリガーしてもよい。

【0024】トレース機能がトリガーされた場合、デバッグはプログラムに対応するアイテムの現在の状態に関連し、かつ該プログラムの実行中に生ずるトレース情報を収集する。この情報は、開発者にとって利用可能となり、この情報を用いてプログラムのデバッグが試みられる。本発明を理解する上で、トレース機能が実行またはトリガーされた場合にデバッグにおいて何が起きているかの詳細は不必要であり、また当業者にとって既知のことである。したがって、本明細書ではそのことについての詳細な説明は行わない。

【0025】本発明のアーキテクチャおよび一般的な論理フローを図2に示す。オブジェクト指向プログラムでは、各オブジェクトに対応するメソッドはプログラムの論理部分を有する。各メソッドの実行にともなう、各メソッドは他のオブジェクトへメッセージを送ることによって、またインスタンス変数のデータを処理することによって論理を実行する。この記載では、インスタンス変数へ"アクセス"する試みは、インスタンス変数を読み取ること、インスタンス変数を修正すること、インスタンス変数を記述すること等の試みを意味することができる。図2はオブジェクト指向プログラム40の一部分を示す。このオブジェクト指向プログラム40は、メモリに存在し、かつオブジェクト44、46から構成される。また、オブジェクト44、46の各々は、オブジェクト44、46の論理部分と該オブジェクト44、46に対応したインスタンス変数52を含むデータ部分50とを構成する多数のメソッド48を含む。修正に先立って、オブジェクト46はデータ部分50を含む点のみがオブジェクト44と異なる。このデータ部分50は開発者によって指定されていたインスタンス変数54を含むもので、トレーシングに影響を及ぼす。本発明によれば、オブジェクト46の指定されたインスタンス変数54に対するいかなるアクセスの試みもアクセス・メソッド56、58のいずれか一つを経由しなければならないように、読み取りおよび書き込みアクセス・メソッド56、58をオブジェクト46に追加する。開発者が設定したパラメータによっては、指定されたインスタンス変数54にアクセスする試みがなされた場合に読み取りアクセス・メソッド56または書き込みアクセス・メソッドのいずれか一方が実行(ふさわしいものとして)される。それによって、トレーシングを実行するためにトレース・メッセージがデバッグ62のトレース・マネージャ60に送られ(また、メモリ42内に論理的に置かれる)、またオブジェクト46の指定されたインスタンス変数54に関連する所望のトレーシングが実行される。

【0026】メソッド56、58によって、実行中に指定インスタンス変数54に対して読み取りアクセスおよび書き込みアクセスがなされる。そして、終了実行に先

立って、指定命令変数54に対する適切なアクセスを行った後にメソッド56、58がさらにトレーシングを引き起こしてもよい。このことは、トレーシングの終了、トレーシング・パラメータまたはどのような情報が収集されるかの修正、あるいはトレーシングの開始等を行うために、第2のトレース・メッセージをトレース・マネージャ60に送ることを含むものであってもよい。メソッド56、58は、アクセス数およびアクセスの種類を数えてもよく、ある種のアクセスあるいは指定インスタンス命令54へのアクセスが一定の回数なされたかどうかによって、あるいは一定の種類のアクセスがトレースの開始および終了がなされてもよい。あるいは、この論理がトレース・マネージャ60に含まれてもよく、アクセスの試みが行われ、かつ該アクセスが生ずる。

【0027】トレース・マネージャ60は、所望のパラメータにもとづいてトレース情報の収集を行い、つづいて同時に既知の方法にもとづいてトレース情報の出力を行う。

【0028】以下、図3のフローチャートを参照しながら、好ましい実施の形態にもとづいてインスタンス変数に関連するトレースポイントの設定に関連した論理を説明する。開発者が要求するインスタンス変数にアクセスすることによるトレースのトリガーあるいはトレースへ影響を与えることに関連して、デバッグ62は開発者が1つ以上のインスタンス変数を指定することを可能にする(ステップ100)。図4は、ソフトウェア・デバッグ・ツールのユーザ・インタフェースに表示されたウィンドウ70を示すものである。ウィンドウ70に設けられたダイアログを通じて、開発者はデバッグしているオブジェクト指向プログラム内のクラスまたはオブジェクトに関連している特定のデータ構成要素またはインスタンス変数に対してトレースポイントを設定することができる。ウィンドウ70には、データ構成要素またはインスタンス変数からなるスクロール可能なリスト70が表示される。同様に、このリスト72内の各データ・アイテムに論理的に関連している多数の選択可能なボックス74もウィンドウ70に表示される。開発者は所望のインスタンス変数に対応したボックス74を選択することによって、リスト72から与えられたインスタンス変数に関連するトレースポイントを設定する。

【0029】図5は、ソフトウェア・ツールのユーザ・インタフェースに表示可能なウィンドウ80を示すもので、選択されたインスタンス変数に関係して設定されているトレースに有効なオプションを開発者に提供する。例えば、開発者は、オブジェクト指向プログラムの実行中において、どのインスタンス変数(すなわち、第1、第10、第18)へのアクセスが、ボックス82の数値を設定することによってターン・オンされるトレース機能を生ずるかを特定することができる。それに加えて、開発者はボックス84の数値を設定することによってト

レーシング機能をターン・オフにする特定のアクセスを指定してもよい。さらに、開発者はアクセスの種類を特定したり、あるいはインスタンス変数に関するトレーシングをトリガーする一定の条件を特定することができる。一定の条件は開発者によって識別されるもので、ボックス86で適当な条件を設定することによってトレース機能が引き起こされる。

【0030】開発者はボックス88を用いることによって、指定されたインスタンス変数のアクセスに先立って、アクセス・メソッド56、57のいずれか一つの実行中に起動または終了するトレーシングを引き起こす。さらに、ボックス90において対応する選択肢を選択することによって、指定されたインスタンス変数54を適当なアクセス・メソッド56、58がアクセスした後に、開発者はトレーシングを停止または開始することができ、あるいは有効と思われる適当なオプションをもたらすことができる。

【0031】ここで再び図3を参照しながら説明する。開発者がどのトレーシングの引き起こしが要求されるかに関連する一つ以上のインスタンス変数を指定（ステップ100）した後、開発者は図5のウィンドウを介してステップ102においてトレース機能の特定の機能パラメータを指定する。この特定の機能パラメータは指定されたインスタンス変数のすべてに関して同一のものであってもよい。つぎに、デバッガ62はプログラムを以下のように修正する。各指定されたインスタンス変数に関連してトレースポイントがインストールされるまで（ステップ104）、指定されたインスタンス変数の一つが選択され、かつその対応するトレースポイントがインストレーション用を選択される。決定は、インスタンス変数の読み取りアクセスにトレーシングが必要かどうかに応じて、まずなされる（ステップ106）。もしそのようであれば、読み取りアクセス・トレースポイントが作られ、さらに指定されたインスタンス変数に関するインタセプト・メッセージに論理的に位置するようにして適当なクラスにインストールされる。図2はインスタンス変数54が位置しているオブジェクト46内に読み取りアクセス・トレースポイント・メソッド56が位置しているようにして説明するためにものである。プログラム40ないの指定されたインスタンス変数54のすべて読み取りアクセスを読み取りアクセス・トレースポイント・メソッド56へ送られるメッセージに置き換える（ステップ110）。したがって、デバ깅する間のプログラム40の実行中に、インスタンス変数54を読み取るすべての試みは、読み取りアクセス・トレース・ポイント・メソッド56を介して行われよう。

【0032】ステップ106で、あるいはステップ110を完了した後で読み取りアクセスに関連する、要求されたトレーシングが存在しないと判断された場合、処理はステップ112へ進む。このステップ112では、開

発者は指定されたインスタンス変数の書き込みアクセスを要求してトレーシングに影響を及ぼす。もしそのようであれば、指定されたインスタンス変数に関連する書き込みアクセス・トレース・ポイント・メソッド58が生成され、オブジェクト46（またはオブジェクト46に関連する適当なオブジェクトまたはクラス）に加えられる（ステップ114）。次に、プログラム40における指定されたインスタンス変数のすべての書き込みアクセスが書き込みアクセス・トレースポイント・メソッド58に送られたメッセージに置き換えられる（ステップ116）。バ깅の間のプログラム40の実行中に、指定されたインスタンス変数54へ書き込みする試みのすべてが書き込みアクセス・メソッド58を経由するであろう。

【0033】指定されたインスタンス変数54の書き込みアクセスに関連したトレーシングの要求がないとステップ112で判断された場合、あるいはステップ116で書き込みアクセス・メソッドのインストレーションが完了した後、処理はステップ104に戻る。ステップ104で、トレーシングをセットアップするための追加の指定インスタンス変数が存在すると判断された場合、それらのインスタンス変数の一つが選択され、選択されたインスタンス変数に関連してステップ106～116が繰り返される。ステップ104において、設定すべき追加のトレースポイントがないと最終判断された場合、セットアップ・プロセスが終了する。

【0034】いくつかのプログラムでは、与えられたインスタンス変数の読み取りおよび書き込みアクセスが既にアクセス・メソッドを経由していてもよい。そのような場合、上記のようにして生成される新規のメソッド、および既存の読み取りまたはアクセス・メソッドの指定されたデータ構成要素へ送られるメッセージが新しく生成されたアクセス・メソッドに言及して適切に修正されるであろう。

【0035】好ましい実施態様では、プログラム40内のメソッドの一つが指定されたインスタンス変数54の読み取りまたは書き込みを試みるたびに、開発者が入力するトレース機能パラメータにもとづいて、トレーシングが開始または終了するかどうかを判断する論理が読み取りおよび書き込みアクセス・トレースポイント・メソッド56、58に含まれる。この読み取りおよび書き込みアクセス・トレースポイント・メソッド56、58は、開発者が入力したトレース機能パラメータを適切に実行させる論理を含む。このトレースポイント・アクセス・メソッド56、58はトレース・メッセージをトレース・マネージャ60へ送り、指定されたパラメータにもとづいてトレーシングを開始させる。あるいは、該パラメータに関連した論理は、処理を行うためにトレース・マネージャ60へメッセージを単純に送るアクセス・メソッド56、58とともに、トレース・マネージャ6

0および(または)デバッガ62に保持してもよい。

【0036】デバ깅中におけるオブジェクト指向プログラムおよびデバ깅・ツール62の実行に関連する論理を、図6を参照しながら以下詳細に説明する。デバ깅の間、デバ깅を受けているオブジェクト指向アプリケーション内のメソッドをデバ깅・ツール62の制御下で実行し、実際の使用に近似させてオブジェクト指向プログラム40を実行する。各々のメソッドの実行(ステップ200)に伴い、実行されているメソッドが読み取りまたは書き込みトレースポイント・アクセス・メソッドのいずれかの一つかどうかを決定する(ステップ202)。そうではない場合、プロセスは正常に進み、さらにメソッドが完了すると実行される次のメソッドの決定および実行が行われる(ステップ204、200)。しかし、ステップ202において、実行されているメソッドがトレースポイント・アクセス・メソッドの一つであると判断された場合、トレース・マネージャ60およびデバッガ62とともにアクセス・メソッドに含まれた論理に基づいて、以下のよう10に進行する。以下のプロセスはアクセス・メソッドの実行によって行われる。実行されているメソッドがアクセス・メソッドではない場合、トレーシングのための以下のプロセスは行われない。このことは行われているメソッドがアクセス・メソッドである状況下でのみトレースに関連してプロセスが行われるので、トレーシングに関連するプロセッサ・オーバーヘッドが最小限に保たれる。

【0037】ステップ206では、開発者によって設定されたパラメータにしたがってアクセスの種類および現在の状態がトレーシングをトリガーするか、もしくはさもなくばトレーシングに影響を与えるかをアクセス・メソッド(またはトレース・マネージャ60)の論理が判断する。もしそうではないと判断された場合、アクセス・メソッドは指定インスタンス変数54への適当なアクセスを可能とし、該アクセス・メソッドはインスタンス変数の記述(格納)あるいはインスタンス変数の読み取り(応答)を適当に行う(ステップ208)。さらに、アクセス・メソッドは非活動状態となり、プログラム全体の実行が次のメソッドの実行によって継続する(ステップ204、200)。

【0038】しかし、ステップ206でアクセス・メソッドの実行中に試みられているアクセスの種類がトレーシングに影響を及ぼすように設定されていると判断された場合、および(または)アクセスに関連した状態がトレーシングに影響を及ぼすのに十分であると判断された場合、プロセスはステップ20へ進み、指定されたパラメータにもとづいてトレーシングが影響を受ける。例えば、トレーシングが開始されるか、もしくはトレーシングが終了することも可能である。このことは、適当なメッセージをトレース・マネージャ60へ送るアクセス・メソッドによって達成される。このアクセス・メソッド

は、既知の方法にもとづいてデバ깅・ツールとともにトレース機能を実行する。

【0039】ステップ210において、アクセスの種類が異なるとそれに応じてトレーシングのトリガーに異なる影響が与えられるようにしてもよい。また、指定インスタンス変数54に含まれる値を読み取るアクセスがトレーシングをトリガーし、一方で指定インスタンス変数54へ値を書き込むアクセスがトリガーされないようにしてもよい。同様に、第1または第5のアクセスがトレーシングをトリガーしないようにする一方で、第10回目のアクセスがトレーシングをトリガーするようにし、さらに12回目のアクセスによってトレーシングが終了するようにしてもよい。このこともまた、開発者によってウィンドウ80を介して設定することができる。上記のことを実現する論理は、アクセスの計数および決定を可能とし、トレース・マネージャ60へメッセージを送り、ユーザ指定パラメータに依存したトレーシングのターン・オンまたはターン・オフを行うように、アクセス・メソッド56、58を生成することによって該アクセス・メソッド56、57内に保持することができよう。あるいは、実行中にトレース・マネージャへメッセージを単に送るアクセス・メソッド56、58によって、この論理をトレース・マネージャ60に含有させることができよう。後者のトレース・メッセージは単にアクセスの種類に関する情報を含むことができ、あるいは代わりに親メッセージからのオリジナル・メッセージとすることもできよう。それによって、指定パラメータを実現する論理は制御トレーシングに最適なメッセージを解釈するトレース・マネージャ60に含まれよう。

【0040】アクセス・メソッドが書き込みアクセス・メソッドである場合、書き込みアクセス・メソッド56は指定インスタンス変数54内の指定インスタンス変数54に書き込もうとするメソッドから受けたメッセージ内に受けた値を書き込む(ステップ212、214)。アクセス・メソッドが読み取りアクセス・メソッドである場合、該アクセス・メソッドは指定インスタンス変数54を読み取ろうとしたプログラム40内のメソッドからメッセージを受け取るであろう。セット・アップ処理の間、指定インスタンス変数54に対する参照は、指定インスタンス変数54を読み取る試みは読み取りアクセス・メソッド58に送られるように、リード・アクセス・メソッド58に対する参照に置き換えられる。ステップ216により、読み取りアクセス・メソッドは親メソッドに代わって指定インスタンス変数54を読み取る。

【0041】ステップ214または216のいずれかの後に、指定インスタンス変数54への適当なアクセスに続いて、読み取りまたは書き込みアクセス・メソッド56、57の実行が再びトレーシングに影響を及ぼすようにしてもよい。図5のボックス90を介して開発者によって設定されたパラメータに依存して、トレーシングが

開始、終了、または設定条件に依存して、あるいは設定条件の発生に対して復元されてもよい(ステップ218)。ステップ218はステップ210に関連する記載と同様な方法で実行されるもので、適当な論理はアクセス・メソッド56、58に含まれ(または代わりに、トレース・マネージャ60)、また適当なメッセージがトレース・マネージャ60に送られてトレーシングに対して適当な影響を及ぼす。

【0042】アクセス・メソッドの実行が完了すると、プログラム40のプロセスは次のメソッドによって継続する(ステップ204、200)。

【0043】注目すべきことは、機能を他の方法でもって実現する。Smalltalkでは、Smalltalkバーチャル・マシンの修正によってよりいっそう容易に機能の実行を行うことができ、プログラムを検索する必要性を不要にするとともに、各インスタンス変数を生成されたアクセス・メソッドへ送られたメッセージによって置き換える。

【0044】主発明を実施する上で必要とされる知識または技術は当業者に十分知れたものである。本発明の一実施形態例が記載された一方で、種々の変更および変形例は本発明の精神および範囲から逸脱することなく可能である。

【0045】まとめとして、本発明の構成に関して以下の事項を開示する。

(1) オブジェクト指向プログラムのデータ構成要素に対するアクセスに基づく、前記オブジェクト指向プログラムのデバッグ中にトレース情報を収集することを可能とするデータ処理環境でのプロセッサ上での実行のためのコンピュータ読み取り可能コードであって、どのデータ構成要素に対するアクセスがトレース情報の収集を開始するのに必要であるかに関連する前記オブジェクト指向プログラムのデータ構成要素を、ユーザが指定することを可能とする第1のサブプロセスと、指定されたデータ構成要素へ送られるメッセージを送ることによってトリガーされたトレース情報の収集を開始する第2のサブプロセスとを備えることを特徴とするコンピュータ読み取り可能コード。

(2) 前記第1のサブプロセスは、前記指定されたデータ構成要素にアクセスすることに基づいた前記トレース情報の回収を制御するパラメータを、さらにユーザが指定できるようにすることを特徴とする上記(1)に記載のコンピュータ読み取り可能コード。

(3) 前記パラメータの一つは、どのトレースされた情報の収集が終了するかに対する事像であり、さらに前記終了の事像の発生に対するトレース情報の回収を終らせる第3のサブプロセスを備えることを特徴とする上記(2)に記載のコンピュータ読み取り可能コード。

(4) 前記第1のサブプロセスは、前記指定されたデータ構成要素に対する少なくとも一つのアクセス・メソッ

ドを生成することと、前記オブジェクト指向プログラムのメソッドにある前記指定されたデータ構成要素に対する直接参照のすべてを、前記アクセス・メソッドに送られるメッセージに置き換えることとを、さらに含むことを特徴とする上記(1)に記載のコンピュータ読み取り可能コード。

(5) 前記指定されたデータ構成要素は、オブジェクト指向プログラムのオブジェクトのインスタンス変数であることを特徴とする上記(1)に記載のコンピュータ読み取り可能コード。

(6) 前記アクセス・メソッドは、トレース情報の収集を制御することを特徴とする上記(4)に記載のコンピュータ読み取り可能コード。

(7) 前記第1のサブプロセスは、読み取りトレースポイント・アクセス・メソッドおよび書き込みアクセス・トレースポイントを生成し、さらに前記オブジェクト指向プログラムによって発生した前記指定されたデータ構成要素に対する読み取りの試みのすべては、読み取りトレースポイント・アクセス・メソッドに送

られ、また前記オブジェクト指向プログラムによって発生した前記指定されたデータ構成要素に対する書き込みの試みのすべては、書き込みトレースポイント・アクセス・メソッドに送られることを特徴とする上記(1)に記載のコンピュータ読み取り可能コード。

(8) 前記アクセス・メソッドは、前記指定されたデータ構成要素にアクセスするメッセージを受け取った後に前記トレース情報の収集を開始させることを特徴とする上記(4)に記載のコンピュータ読み取り可能コード。

(9) アクセス・メッセージを受け取った後に要求されたアクセス機能を実行する前記アクセス・メソッドを生成する第4のサブプロセスと、前記アクセス・メソッドが前記要求されたアクセス機能を実行した後に、トレース・データの収集を修正する第5のサブプロセスとを、さらに備えることを特徴とする上記(8)に記載のコンピュータ読み取り可能コード。

(10) 少なくともプロセッサとメモリとが含まれるコンピュータ環境下で、デバッグを受けているオブジェクト指向プログラムのデータ構成要素へのアクセスに基づいて収集されるべきトレース・データを生ずるシステムであって、前記データ構成要素へのアクセスの試みがトレース情報の収集を開始するために要求される前記オブジェクト指向プログラムの少なくとも一つのデータ構成要素を指定する手段と、指定された前記データ構成要素がアクセスされた場合に、トレース・データの収集をトリガーする手段とを備えたことを特徴とするシステム。

(11) トレーシング・パラメータを指定する手段を、さらに有することを特徴とする上記(10)に記載のシステム。

(12) デバッグしているオブジェクト指向プログ

ラムのデータ構成要素がアクセスされた場合のトレース・データ収集方法であって、アクセスの試みがトレーシングに影響を及ぼす前記オブジェクト指向プログラムのデータ構成要素を指定する段階と、指定されたデータ構成要素に対するアクセスの試みがデバッグ中に送られる少なくとも一つのアクセス・メソッドを生成する段階と、前記アクセス・メソッドが前記指定されたデータ構成要素に対するアクセスの試みを受けた場合にトレース・データの収集をトリガーする段階とを有することを特徴とするトレース・データ収集方法。

(13) 前記少なくとも一つのアクセス・メソッドを生成する段階は、前記指定されたデータ構成要素を要求するすべての読み取りアクセスと、前記指定されたデータ構成要素を要求するすべての書き込みアクセスとを生成することを特徴とする上記(12)に記載のトレース・データ収集方法。(14) 前記アクセス・メソッドが前記アクセスの試みを受けた後、前記指定されたデータ構成要素にアクセスする段階と、前記指定されたデータ構成要素がアクセスされた後に前記トレース・データの収集を修正する段階とを、さらに有することを特徴とする上記(12)に記載のトレース・データ収集方法。

【図面の簡単な説明】

【図1】本発明が適用されるハードウェア環境の概略的構成を示すブロック図である。

【図2】本発明を利用する一方で、デバッグされているプログラムの実行中におけるデータの論理フローを概略的に説明する模式的構成図である。

【図3】本発明にもとづいて、データ構成要素のトレースポイントを設定することに関連する論理段階を説明するためのフローチャートである。

【図4】デバッグされるオブジェクト指向プログラムの特定のデータ構成要素に対して開発者がテストポイントを設定することが可能なデバッグ・ツールまたはルーチンに使用されるウィンドウを説明するための模式図である。

【図5】開発者が特定のトレースポイント・パラメータを設定することが可能となるデバッグ・ツールまた

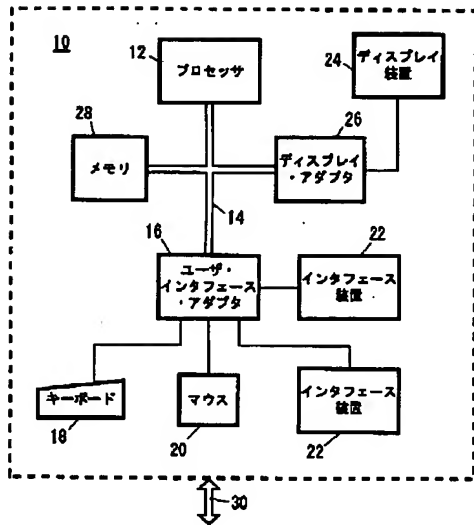
はルーチンに使用されるウィンドウを説明するための模式図である。

【図6】オブジェクト指向プログラムのデータ構成要素へのアクセスに基づいたトレース機能をトリガーする論理段階を説明するためのフローチャートである。

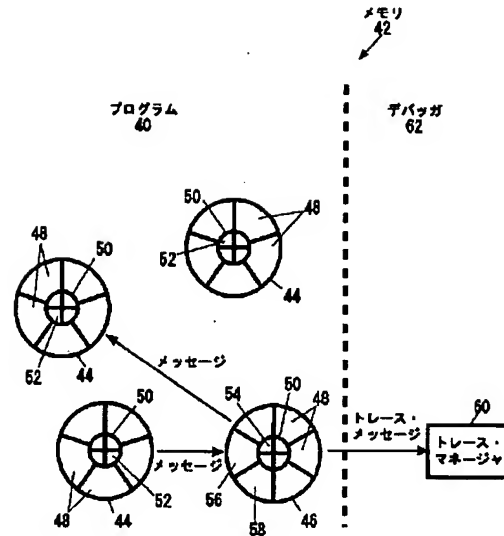
【符号の説明】

10	ワークステーション
12	プロセッサ
14	バス
16	ユーザ・インタフェース・アダプタ
18	キーボード
20	マウス
22	インタフェース装置
24	ディスプレイ装置
26	ディスプレイ・アダプタ
28	メモリ
30	通信チャンネル
40	プログラム
42	メモリ
44	オブジェクト
46	オブジェクト
48	メソッド
50	データ部分
52	インスタンス変数
54	インスタンス変数
56	アクセス・メソッド
58	アクセス・メソッド
60	トレース・マネージャ
62	デバッグ
70	ウィンドウ
72	リスト
74	ボックス
80	ウィンドウ
82	ボックス
84	ボックス
86	ボックス
88	ボックス

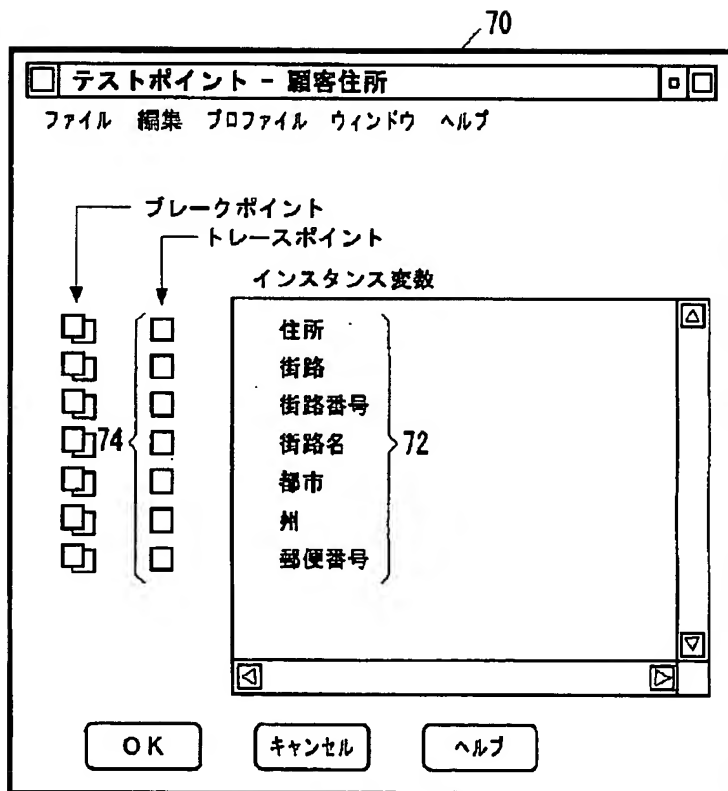
【図1】



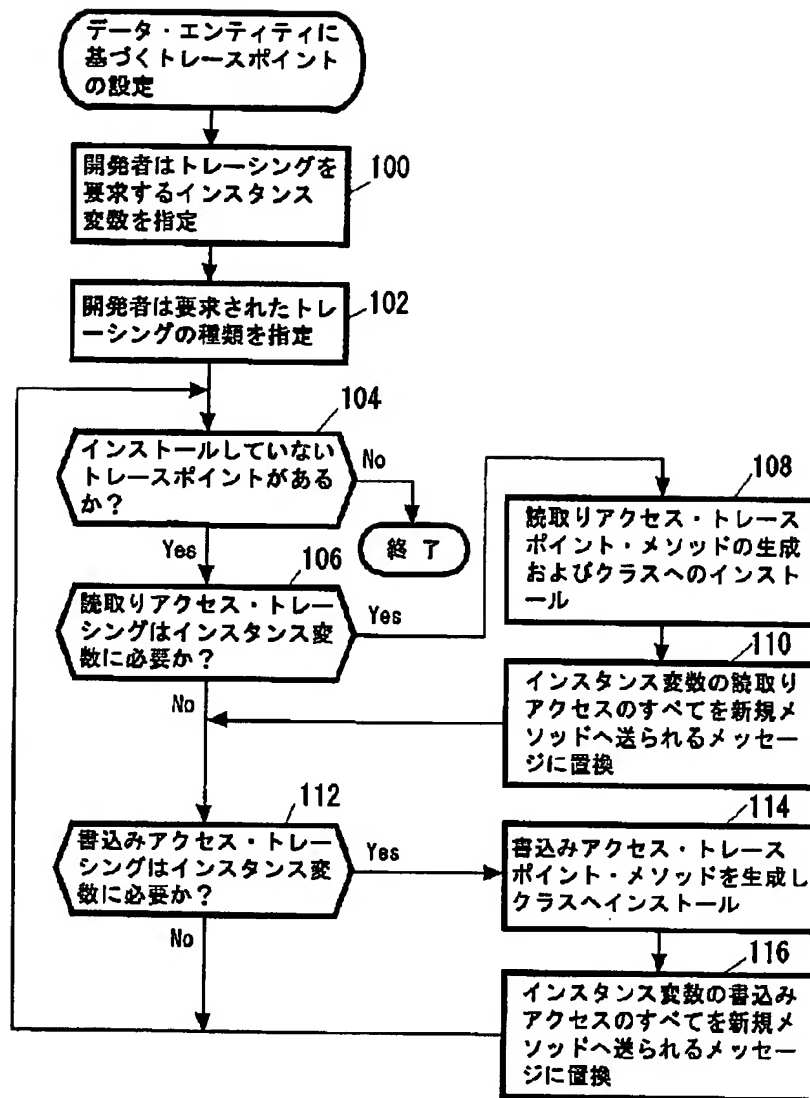
【図2】



【図4】



【図3】



【図5】

80

条件付きトレースポイント

トレースオプションの状況

エンタウンタから 82 84 (1.65535)

エンタウンタへ 85 86 (1.65535)

1 条件

アクセス前のトレースオプション 88

☒ トレース・オン ☐ トレース・オフ

アクセス後のトレースオプション

☐ トレース・オン ☐ トレース・オフ ☒ トレース・オプションの回復 90

OK キャンセル ヘルプ

【図6】

